

Week 12 - Hashing

- 주의 사항: 부정행위 금지(채점서버 외 인터넷 사용금지), STL 사용금지 (string, vector는 사용 가능)
- 표준 입출력 사용을 권장 (C는 scanf / printf, C++은 cin / cout)

문제 1 (Easy version)

N 개의 자연수로 이루어진 해시 테이블을 작성하는 프로그램을 만들어보자. 충돌이 일어나는 경우, Linear probing을 사용하여 테이블에 추가한다. 해시 함수는 다음의 해시 함수를 사용한다.

```
#include<iostream>

#define MAX 353333

class item {
public:
    int key, value;
};

class arr {
    item* data;
    bool flag = false;
};

arr HashArr[MAX];
int sz = 0;

int hashfunc(int idx) {
    return idx % MAX;
}
```

입력

- 1. 첫 번째 줄에는 테스트케이스의 수 T 가 주어진다. ($1 \leq T \leq 1,000$)
(아래의 과정이 T 회 반복된다.)
- 2. 두 번째 줄에는 해시 테이블을 구성할 자연수의 수 N 이 주어진다. ($1 \leq N \leq 1,000$)
- 3. 세 번째 줄에는 해시 테이블을 구성할 중복되지 않는 N 개의 자연수가 빈칸을 사이에 두고 주어진다.
- 4. 네 번째 줄에는 해시 테이블에서 탐색할 자연수의 수 M 이 주어진다. ($1 \leq M \leq N$)
- 5. 다섯 번째 줄에는 해시 테이블에서 탐색할 M 개의 자연수가 빈칸을 사이에 두고 주어진다.

출력

탐색할 M 개의 자연수에 대하여, 주어진 입력이 해시 테이블에 존재하면 1과 probing 횟수를, 존재하지 않으면 0과 probing 횟수를 빈칸을 사이에 두고 출력한다. 충돌이 없는 경우를 probing 1회라고 정의한다.

예제 입출력

예제 입력	예제 출력
2	1 1
5	1 1
4 1 5 2 3	0 1
5	0 1
1 3 7 9 5	1 1
6	1 4
1 353334 706667 2 353335 3	1 1
6	1 3
3 1 706667 2 353334 353335	1 3
	1 2
	1 4